



PROGETTO INGEGNERIA DEL SOFTWARE

Parte 3 - Java

Abiuso Matteo

Barbagiovanni Minciullo Carlo

Consonni Luca



SOMMARIO

- Struttura base Progetto:

- Server

- COM.GROUP15.SERVER
- COM.GROUP15.GESTORI/COM.GROUP15.INTERFACECOM.GRUOP15.DATA
- COM.GROUP15.TEST

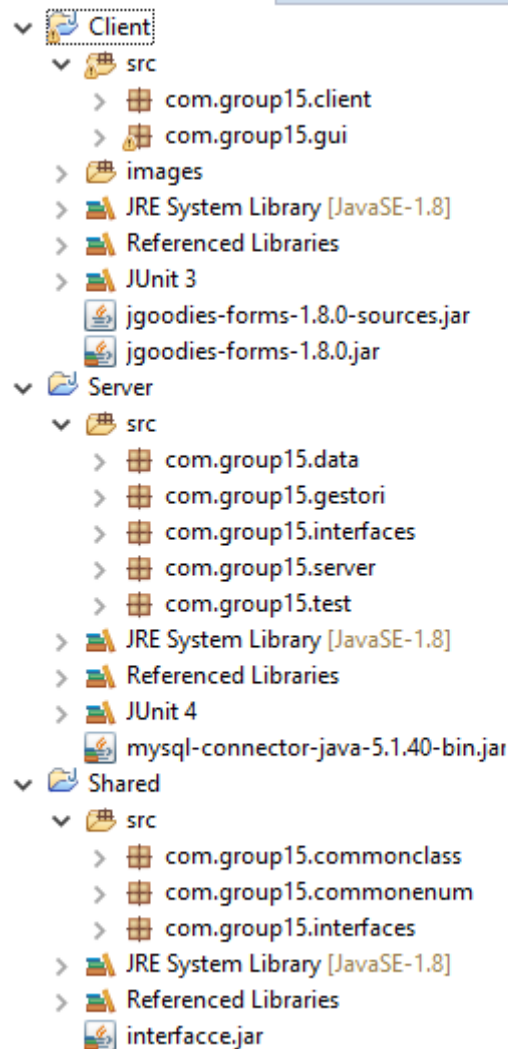
- Client

- COM.GROUP15.CLIENT
- COM.GROUP15.GUI

- SQL

STRUTTURA BASE PROGETTO

DIVISIONE IN 3 PROGETTI



Per l'implementazione del programma abbiamo dovuto suddividere in 3 sottoprogetti il nostro lavoro, come si può vedere in figura.

Il primo progetto è il server, svolge la funzione al database e relativi metodi. E' suddiviso in 5 packages: server, data, gestori, interfaces e test.

Il secondo è il client, dove risiede l'interfaccia grafica (.gui) e i metodi con cui la persona fisica interagisce con il nostro software.

Mentre lo Shared, ci permette di fare comunicare tra loro Client e Server, mettendo a disposizione classi, enum e interfacce comuni.

SERVER

COM.GRUOP15.SERVER

In questo packages troviamo la classe Server.java nel quale risiede il main di questo progetto. Come prima operazione viene inizializzato il server e i relativi gestori, dopodichè, sempre all'interno di un try/catch tenta la connessione al database. Il server implementa l'interfaccia: Server Interface, di cui è un façade, ovvero un pattern che permette, attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce complesse e molto diverse tra loro, nonché a blocchi di codice complessi, rappresentate in questo caso dalle interfacce dei relativi gestori (qui sotto uno stralcio).

```
@Override
public boolean checkLogin(String username, String password) throws RemoteException {
    return gestoreAccount.checkLogin(username, password);
}

@Override
public void addAccount(Account a) throws RemoteException {
    gestoreAccount.addAccount(a);
}

@Override
public void deleteAccount(String username) throws RemoteException {
    gestoreAccount.deleteAccount(username);
}
```

COM.GRUOP15.GESTORI/COM.GROUP15.INTERFACE

In questi due packages vi sono le classi gestori e le relative interfacce.

I gestori raccolgono i metodi delle rispettive categorie account, cliente e progetto, che andranno a chiamare le funzione di interazione diretta con il database.

```

package com.group15.gestori;

import java.rmi.RemoteException;

public class GestoreCliente extends UnicastRemoteObject implements ClientiInterface{

    private static final long serialVersionUID = 6102431945702637135L;

    public GestoreCliente() throws RemoteException {
        super();
        // TODO Auto-generated constructor stub
    }

    @Override
    public void addCliente(Cliente c) throws RemoteException {
        try {
            DatabaseManager.getInstance().addCliente(c);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

1 package com.group15.interfaces;
2
3 import java.rmi.RemoteException;
4
5 public interface ClientiInterface {
6
7     public void addCliente(Cliente c) throws RemoteException;
8     public void deleteCliente(String email) throws RemoteException;
9     public Vector<Cliente> getClienti() throws RemoteException;
10
11 }

```

COM.GRUOP15.DATA

In questo package abbiamo la classe DatabaseManager.java. Come prima cosa viene istanziato un singleton, che ha lo scopo di garantire che della determinata classe venga creata una e una sola istanza, e di fornire un punto di accesso globale. Inoltre i metodi di quella classe comunicano direttamente con il database, creando statement e eseguendo queries.

```

import java.sql.Connection;

public class DatabaseManager {

    private static final String URL = "jdbc:mysql://localhost:3306/marketing_solution?autoReconn
    private static final String user = "MsAdmin";
    private static final String password = "MsAdmin";

    private static DatabaseManager instance;
    private Connection connection;

    private DatabaseManager() throws SQLException, ClassNotFoundException{           //Connessione
        Class.forName("com.mysql.jdbc.Driver");
        connection = DriverManager.getConnection(URL, user, password);
        System.out.println("Connesso al database");
    }

    public static DatabaseManager getInstance() throws SQLException, ClassNotFoundException{ //
        if(instance == null)
            instance = new DatabaseManager();
        return instance;
    }

    public void addAccount(Account a) throws SQLException {

        String sql = "INSERT INTO accounts(nome,cognome,username,password) VALUES(?,?,?,?);";
        PreparedStatement stmt = connection.prepareStatement(sql);
        stmt.setString(1, a.getNome());
        stmt.setString(2, a.getCognome());
        stmt.setString(3, a.getUsername());
        stmt.setString(4, a.getPassword());
        stmt.execute();
        stmt.close();
    }
}

```

COM.GROUP15.TEST

In questo package abbiamo la classe DatabaseManagerTest, in cui sempre tramite singleton viene istanziato un nuovo database, tuttavia questa volta vuoto, in cui tramite la classe ServerTest, andremo a testare i metodi principali del nostro software mediante l'utilizzo di JUnit.

```

@Test
public void testGestoreCliente() throws SQLException {
    Cliente cliente = new Cliente(1, "Mario", "Rossi", "mariorossi@hotmail.it", "3331874207");

    //addCliente
    database.addCliente(cliente);

    listaClienti = database.getClienti();
    assertEquals(listaClienti.elementAt(0).getNome(), "Mario");
    assertEquals(listaClienti.elementAt(0).getCognome(), "Rossi");
    assertEquals(listaClienti.elementAt(0).getEmail(), "mariorossi@hotmail.it");
    assertEquals(listaClienti.elementAt(0).getNumero(), "3331874207");

    database.deleteCliente("mariorossi@hotmail.it");
}

```

CLIENT

COM.GRUOP15.CLIENT

All'interno di questo packages troviamo la classe Client.java in cui risiede il main del progetto Client, ovvero ciò che ci permette di interagire con l'utente reale. Qui implementiamo un'interfaccia di tipo ServerInterface in modo tale da raggiungere i metodi del facade. Qui viene lanciata la prima gui (graphic user interface).

```
public static void main(String[] args) throws RemoteException {  
    try {  
        Registry r = LocateRegistry.getRegistry("localhost",1234); // Try per la connessione al server e alla su  
        server = (ServerInterface) r.lookup("MsServer");  
        System.out.println("Client connesso al server");  
    } catch (RemoteException | NotBoundException e) { //Catch per la mancata connessione al server  
        System.err.println("Impossibile connettersi al server!");  
        e.printStackTrace();  
    }  
  
    Login.open(); // Apertura della schermata di login
```

Inoltre all'interno del try/catch tentiamo la connessione al server.

COM.GRUOP15.GUI

Dentro questo packages si trovano tutte le schermata grafiche con cui l'utente interagisce direttamente. Le schermate con prefisso add, ci permettono tramite inserimento dati di aggiungere nuove informazioni che andranno poi nel database, come informazioni riguardanti progetti, clienti e account.

Schermate di tipo show ci permettono di visualizzare a video informazioni già presenti nei nostri database.

Mentre quelle con prefisso Set, andranno a modificare contenuti già esistenti.

Login è la schermata iniziale, mentre HomeMan, la schermata principale che collega tutte le altre.

Nome Progetto:

Tipo di Progetto:

Scadenza

Dipendenti da assegnare:

Prezzo: euro

Nome Progetto: Fiera del Raviolo

Nome Cliente: Mario Rossi

Tipo Progetto: FIERA

Tempo Impiegato: 29.0

Prezzo: 15000.0

StatoOrdine: ACCETTATO

Scadenza: 1 Febbraio 2019

Dipendenti Assegn... 2

Proposta 1: No

Proposta 2: SI

Descrizione: Stand ad angolo, 5m x 8m, doppia entrata, con esposizione a vista

Materiali:

Tipo Materiale	Prezzo
Tavoli	350.0

Fatturato: 14186.0 euro

Imposta tempo impiegato

PREVENTIVATO ▼

Annulla Fine

SQL

Nel nostro database: marketing_solution, oltre alle tabelle “base” account e cliente, abbiamo deciso di implementare diverse tabelle che vanno a specificare ogni tipo di progetto, invece di un'unica sola tabella contenente tutti i progetti.

- ▶ clients
- ▶ materiali
- ▶ progetticartellonistica
- ▶ progettievento
- ▶ progettifiera
- ▶ progettionline
- ▶ progettistampa
- ▶ serviziesterni